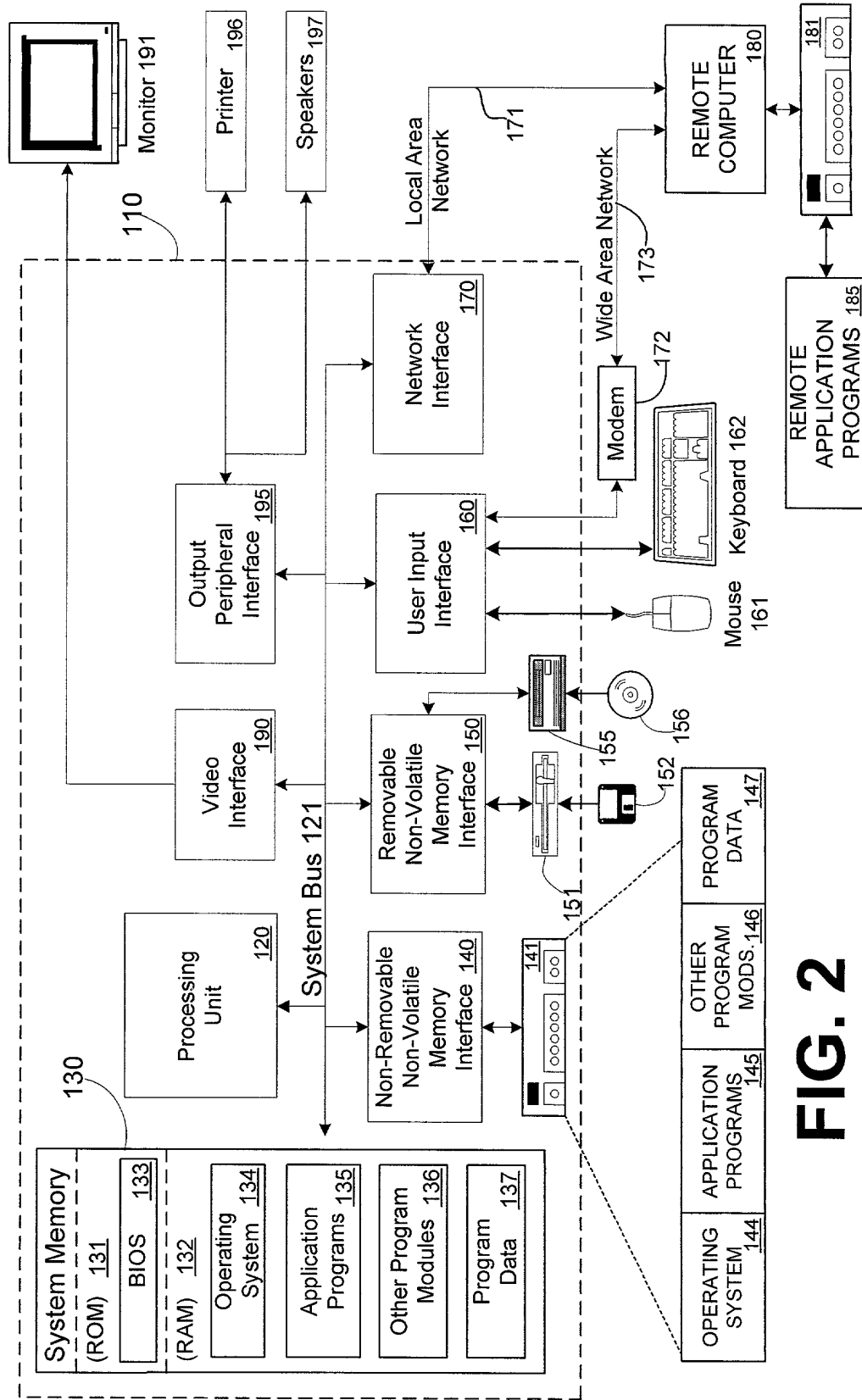


## Computing Environment 100



**FIG. 2**

## FIG. 3A

300

```
<Envelope xmlns="http://schemas.xmlsoap.org/soap/envelope/">
  <Header>
    <!-- user data -->
  </Header>
  <Body>
    <!-- user data -->
  </Body>
</Envelope>
```

## FIG. 3B

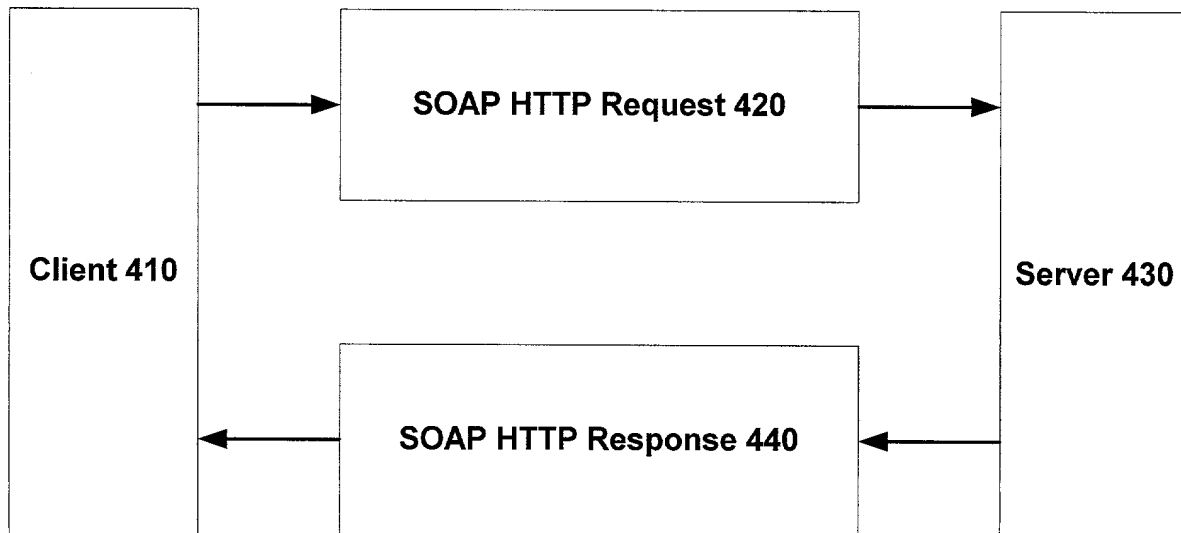
305

```
<SOAP-ENV:Header>
<t:Transaction
  xmlns:t="some-URI" SOAP-ENV:mustUnderstand="1">
  5
</t:Transaction>
</SOAP-ENV:Header>
```

**FIG. 4A**

400

BSTR HelloWorld(BSTR inputString)



**FIG. 4B**

## FIG. 4C

450

POST /HelloWorld HTTP/1.1

Host: <someserver>

Content-Type: text/xml

Content-Length: nnnn

SOAPAction: "Some-URI"

```
<Envelope xmlns="http://schemas.xmlsoap.org/soap/envelope/">
  <Body>
    <HelloWorld xmlns="Some-URI">
      <inputString>StringValue</inputString>
    </HelloWorld>
  </Body>
</Envelope>
```

## FIG. 4D

455

HTTP/1.1 200 OK

Content-Type: text/xml

Content-Length: nnnn

```
<Envelope xmlns="http://schemas.xmlsoap.org/soap/envelope/">
  <Body>
    <HelloWorldResponse xmlns="Some-URI">
      <return>Hello StringValue</return>
    </HelloWorldResponse>
  </Body>
</Envelope>
```

## FIG. 4E

460

```
#define _WIN32_WINNT 0x0500
#define _ATL_ATTRIBUTES
#include <atlsoap.h>

[ module(name="HelloWorld") ];

[ uuid("643cd054-24b0-4f34-b4a1-642519836ef9"), object ]
__interface IHelloWorld
{
    [id(1)] HRESULT HelloWorld(
        [in] BSTR inputString,
        [out, retval] BSTR *retval);
};

[
    request_handler(
        name="Default",
        sdl="sdl"),
    soap_handler(
        name="HelloWorld",
        namespace="http://atlserver/soap/HelloWorld",
        protocol="soap")
]
class CHelloWorld : public IHelloWorld
{
public:
    [ soap_method ]
    HRESULT HelloWorld(
        BSTR inputString,
        BSTR *retval)
    {
        CComBSTR bstrRet(L"Hello ");
        bstrRet += inputString;

        *retval = bstrRet.Detach();

        return S_OK;
    }
};
```

**FIG.**  
**5**

## FIG. 6

600

```
[ uuid("643cd054-24b0-4f34-b4a1-642519836fe8"), object ]  
__interface Interface  
{  
    [id(1)] HRESULT method1([in] bool bInput, [out, retval] bool *bOutput);  
};  
  
[ request_handler, soap_handler ]  
class CClass : public Interface  
{  
    [ soap_method ]  
    HRESULT method1(bool bInput, bool *bOutput)  
    {  
        *bOutput = !bInput;  
        return S_OK;  
    }  
};
```



```
struct __CClass_method1_struct
{
    bool bInput;
    bool bOutput;
};
```

700

**FIG. 7A**

```
extern __declspec(selectany) const _soapmapentry
__CClass_method1_entries[] =
{
    {
        0x38998673,
        "bOutput",
        L"bOutput",
        sizeof("bOutput")-1,
        SOAPTYPE_BOOLEAN,
        SOAPFLAG_RETVAL | SOAPFLAG_OUT,
        offsetof(__CClass_method1_struct, bOutput),
        NULL,
        NULL,
        -1,
    },
    {
        0xEA00C812,
        "bInput",
        L"bInput",
        sizeof("bInput")-1,
        SOAPTYPE_BOOLEAN,
        SOAPFLAG_NONE | SOAPFLAG_IN,
        offsetof(__CClass_method1_struct, bInput),
        NULL,
        NULL,
        -1,
    },
    { 0x00000000 }
};
```

705

**FIG.  
7B**

FIG. 7A is a schematic diagram of a SOAP-based web service.

```
extern __declspec(selectany) const _soapmap
__CClass_method1_map =
{
    0xDC7296CC,
    "method1",
    L"method1",
    sizeof("method1")-1,
    sizeof("method1")-1,
    SOAPMAP_FUNC,
    __CClass_method1_entries,
    sizeof(__CClass_method1_struct),
    1,
    0
};
```

710

**FIG. 7C**

```
extern __declspec(selectany) const _soapmapentry
__CClass_method1_atlsoapheader_entries[] =
{
    { 0x00000000 }
};
```

715

**FIG. 7D**

```
extern __declspec(selectany) const _soapmap
__CClass_method1_atlsoapheader_map =
{
    0xDC7296CC,
    "method1",
    L"method1",
    sizeof("method1")-1,
    sizeof("method1")-1,
    SOAPMAP_HEADER,
    __CClass_method1_atlsoapheader_ent
ries,
    0,
    0,
    -1
};
```

720

**FIG. 7E**

```
extern __declspec(selectany) const _soapmap * __CClass_funcs[] =
{
    &__CClass_method1_map,
    NULL
};

extern __declspec(selectany) const _soapmap * __CClass_headers[] =
{
    &__CClass_method1_atlsoapheader_map,
    NULL
};
```

725

**FIG. 7F**

800

```
ATL_NOINLINE inline const _soapmap ** CClass::GetFunctionMap()
{
    return ____CClass_funcs;
};

ATL_NOINLINE inline const _soapmap ** CClass::GetHeaderMap()
{
    return ____CClass_headers;
}

ATL_NOINLINE inline void * CClass::GetHeaderValue()
{
    return this;
}
```

•  
•  
•

**FIG. 8A**

## FIG. 8B

800



```
ATL_NOINLINE inline HRESULT CClass::CallFunction(
    void *pvParam,
    const wchar_t *wszLocalName,
    int cchLocalName,
    size_t nlItem)
{
    wszLocalName;
    cchLocalName;

    HRESULT hr = S_OK;

    switch(nlItem)
    {
    case 0:
    {
        __CClass_method1_struct *p = (__CClass_method1_struct *) pvParam;
        hr = method1(p->bInput, &p->bOutput);
        break;
    }
    default:
        hr = E_FAIL;
    }

    return hr;
}s

ATL_NOINLINE inline const wchar_t * CClass::GetNamespaceUri()
{
    return L"http://atlserver/soap/test";
}

ATL_NOINLINE inline const char * CClass::GetNamespaceUriA()
{
    return "http://atlserver/soap/test";
}

ATL_NOINLINE inline const char * CClass::GetServiceName()
{
    return "SoapTest";
}
```

## FIG. 9

900

```
[ uuid("643cd054-24b0-4f34-b4a1-642519836fe8"), object ]
__interface Interface
{
    [id(1)] HRESULT method1([in] bool bInput, [out, retval] bool
*bOutput);
};

[ request_handler, soap_handler ]
class CClass : public Interface
{
    int Header1;

    [ soap_method ]
    [soap_header(value="Header1", in="true", out="false",
required="false")]
    HRESULT method1(bool bInput, bool *bOutput)
    {
        *bOutput = !bInput;
        return S_OK;
    }
};
```

```
struct __CClass_method1_struct
{
    bool blnput;
    bool bOutput;
};
```

1000

**FIG. 10A**

```
extern __declspec(selectany) const _soapmapentry
__CClass_method1_entries[] =
{
    {
        0x38998673,
        "bOutput",
        L"bOutput",
        sizeof("bOutput")-1,
        SOAPTYPE_BOOLEAN,
        SOAPFLAG_RETVAL | SOAPFLAG_OUT,
        offsetof(__CClass_method1_struct, bOutput),
        NULL,
        NULL,
        -1,
    },
    {
        0xEA00C812,
        "blnput",
        L"blnput",
        sizeof("blnput")-1,
        SOAPTYPE_BOOLEAN,
        SOAPFLAG_NONE | SOAPFLAG_IN,
        offsetof(__CClass_method1_struct, blnput),
        NULL,
        NULL,
        -1,
    },
    { 0x00000000 }
};
```

1005

**FIG.  
10B**

```
extern __declspec(selectany) const _soapmap
__CClass_method1_map =
{
    0xDC7296CC,
    "method1",
    L"method1",
    sizeof("method1")-1,
    sizeof("method1")-1,
    SOAPMAP_FUNC,
    __CClass_method1_entries,
    sizeof(__CClass_method1_struct),
    1,
    0
};
```

1010

**FIG.  
10C**

```
extern __declspec(selectany) const _soapmapentry
__CClass_method1_atlsoapheader_entries[] =
{
    {
        0xFAD45C89,
        "Header1",
        L"Header1",
        sizeof("Header1")-1,
        SOAPTYPE_INT,
        SOAPFLAG_NONE |
SOAPFLAG_IN,
        offsetof(CClass, Header1),
        NULL,
        NULL,
        -1,
    },
    { 0x00000000 }
};
```

1015

**FIG.  
10D**



```
extern __declspec(selectany) const _soapmap
__CClass_method1_atlsoapheader_map =
{
    0xDC7296CC,
    "method1",
    L"method1",
    sizeof("method1")-1,
    sizeof("method1")-1,
    SOAPMAP_HEADER,
    __CClass_method1_atlsoapheader_ent
ries,
    0,
    0,
    -1
};
```

1020

**FIG.  
10E**

```
extern __declspec(selectany) const _soapmap *
__CClass_funcs[] =
{
    &__CClass_method1_map,
    NULL
};

extern __declspec(selectany) const _soapmap *
__CClass_headers[] =
{
    &__CClass_method1_atlsoapheader_map,
    NULL
};
```

1025

**FIG.  
10F**

20060720 16:22:00

1100

```
ATL_NOINLINE inline const _soapmap ** CClass::GetFunctionMap()
{
    return ____CClass_funcs;
};

ATL_NOINLINE inline const _soapmap ** CClass::GetHeaderMap()
{
    return ____CClass_headers;
}

ATL_NOINLINE inline void * CClass::GetHeaderValue()
{
    return this;
}
```

⋮

**FIG.**  
**11A**

## FIG. 11B

1100



```
ATL_NOINLINE inline HRESULT CClass::CallFunction(
    void *pvParam,
    const wchar_t *wszLocalName,
    int cchLocalName,
    size_t nItem)
{
    wszLocalName;
    cchLocalName;

    HRESULT hr = S_OK;

    switch(nItem)
    {
    case 0:
    {
        __CClass_method1_struct *p = (__CClass_method1_struct *) pvParam;
        hr = method1(p->bInput, &p->bOutput);
        break;
    }
    default:
        hr = E_FAIL;
    }

    return hr;
}s

ATL_NOINLINE inline const wchar_t * CClass::GetNamespaceUri()
{
    return L"http://atlserver/soap/test";
}

ATL_NOINLINE inline const char * CClass::GetNamespaceUriA()
{
    return "http://atlserver/soap/test";
}

ATL_NOINLINE inline const char * CClass::GetServiceName()
{
    return "SoapTest";
}
```

## FIG. 12

1200

```
[ export ]
enum TestEnum { Test_Zero, Test_One, Test_Two, Test_Three, Test_Four };

[ export ]
typedef struct
{
    int n;
    double d;
    BSTR s;
    int arr[2][4];
} TestStruct;

[ uuid("643cd054-24b0-4f34-b4a1-642519836fe8"), object ]
__interface Interface
{
    [id(1)] HRESULT method1([in] TestStruct t, [out, retval]
TestEnum *e);
};

[ request_handler, soap_handler ]
class CClass : public Interface
{
    [ soap_method ]
    HRESULT method1(TestStruct , TestEnum *e)
    {
        // user code ...
    }
};
```

**FIG. 13A**

1300

```
__if_not_exists(__TestEnum_entries)
{
    extern __declspec(selectany) const _soapmapentry __TestEnum_entries[] =
    {
        { 0x18AD26DB, "Test_Four", L"Test_Four", sizeof("Test_Four")-1, 4,
        SOAPFLAG_FIELD, 0, NULL, NULL, -1 },
        { 0x2F4B7497, "Test_Three", L"Test_Three", sizeof("Test_Three")-1, 3,
        SOAPFLAG_FIELD, 0, NULL, NULL, -1 },
        { 0xF8FDB9D9, "Test_Two", L"Test_Two", sizeof("Test_Two")-1, 2,
        SOAPFLAG_FIELD, 0, NULL, NULL, -1 },
        { 0xF8FDA361, "Test_One", L"Test_One", sizeof("Test_One")-1, 1,
        SOAPFLAG_FIELD, 0, NULL, NULL, -1 },
        { 0x18B7F37F, "Test_Zero", L"Test_Zero", sizeof("Test_Zero")-1, 0,
        SOAPFLAG_FIELD, 0, NULL, NULL, -1 },
        { 0x00000000 }
    };
};
```

1305

```
extern __declspec(selectany) const _soapmap __TestEnum_map =
{
    0xF8EFE655,
    "TestEnum",
    L"TestEnum",
    sizeof("TestEnum")-1,
    sizeof("TestEnum")-1,
    SOAPMAP_ENUM,
    __TestEnum_entries,
    sizeof(::TestEnum),
    1,
    -1
};
}
```

**FIG. 13B**

1310

```
__if_not_exists(__TestStruct_entries)
{
    extern __declspec(selectany) const int __TestStruct_arr_dims[] = {2, 2, 4, };

    extern __declspec(selectany) const _soapmapentry __TestStruct_entries[] =
    {
        {
            0x0000006E,
            "n",
            L"n",
            sizeof("n")-1,
            SOAPTYPE_INT,
            SOAPFLAG_FIELD,
            offsetof(::TestStruct, n),
            NULL,
            NULL,
            0
        },
        {
            0x00000064,
            "d",
            L"d",
            sizeof("d")-1,
            SOAPTYPE_DOUBLE,
            SOAPFLAG_FIELD,
            offsetof(::TestStruct, d),
            NULL,
            NULL,
            0
        },
    },
```

•  
•  
•

**FIG. 13C**

## FIG. 13D

1310

```
{
    0x00000073,
    "s",
    L"s",
    sizeof("s")-1,
    SOAPTYPE_STRING,
    SOAPFLAG_FIELD,
    offsetof(::TestStruct, s),
    NULL,
    NULL,
    0
},
{
    0x0001ABC5,
    "arr",
    L"arr",
    sizeof("arr")-1,
    SOAPTYPE_INT,
    SOAPFLAG_FIELD | SOAPFLAG_FIXEDARR,
    offsetof(::TestStruct, arr),
    __TestStruct_arr_dims,
    NULL,
    0
},
{ 0x00000000 }
};
```

**FIG. 13E**

1315

```
extern __declspec(selectany) const _soapmap ____TestStruct_map =  
{  
    0x15962585,  
    "TestStruct",  
    L"TestStruct",  
    sizeof("TestStruct")-1,  
    sizeof("TestStruct")-1,  
    SOAPMAP_STRUCT,  
    ____TestStruct_entries,  
    sizeof(::TestStruct),  
    4,  
    -1  
};  
}
```

1320

```
struct ____CClass_method1_struct  
{  
    TestStruct tIn;  
    TestEnum eOut;  
};
```

**FIG. 13F**



```
extern __declspec(selectany) const _soapmapentry
__CClass_method1_entries[] =
{
    {
        0x38998673,
        "eOut",
        L"eOut",
        sizeof("eOut")-1,
        SOAPTYPE_UNK,
        SOAPFLAG_RETVAL | SOAPFLAG_OUT,
        offsetof(__CClass_method1_struct, eOut),
        NULL,
        &__TestEnum_map,
        -1,
    },
    {
        0xEA00C812,
        "bInput",
        L"bInput",
        sizeof("bInput")-1,
        SOAPTYPE_UNK,
        SOAPFLAG_NONE | SOAPFLAG_IN,
        offsetof(__CClass_method1_struct, tIn),
        NULL,
        &__TestStruct_map,
        -1,
    },
    { 0x00000000 }
};
```

1325

**FIG.  
13G**

```
extern __declspec(selectany) const _soapmap __CClass_method1_map =
{
    0xDC7296CC,
    "method1",
    L"method1",
    sizeof("method1")-1,
    sizeof("method1")-1,
    SOAPMAP_FUNC,
    __CClass_method1_entries,
    sizeof(__CClass_method1_struct),
    1,
    0
};
```

1330

**FIG.  
13H**

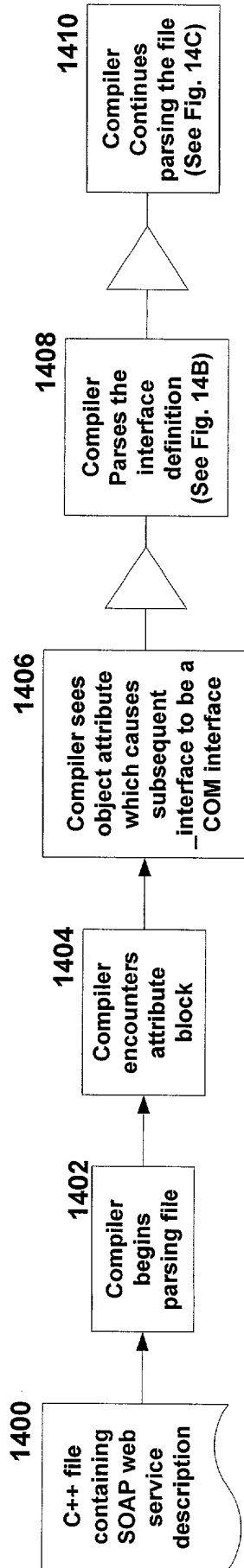


FIG. 14A

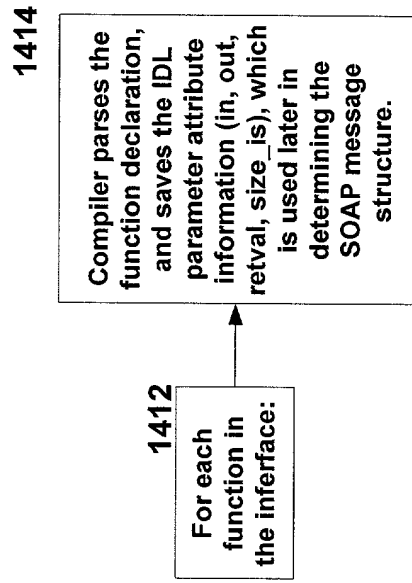
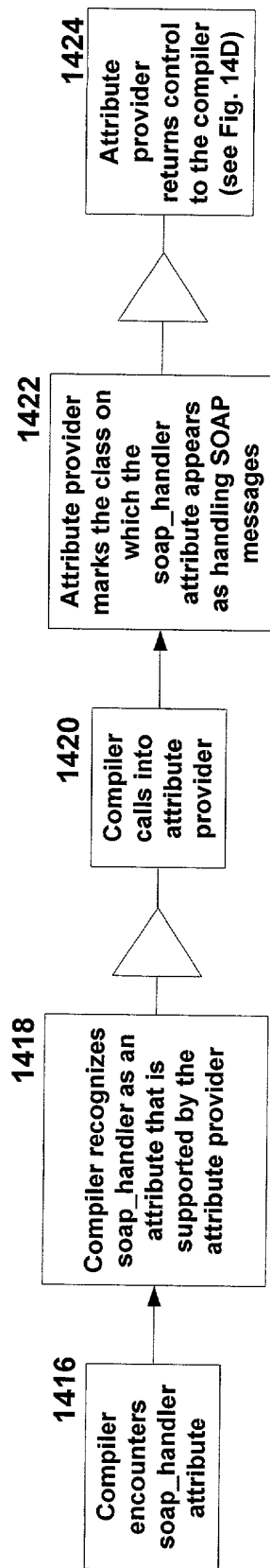
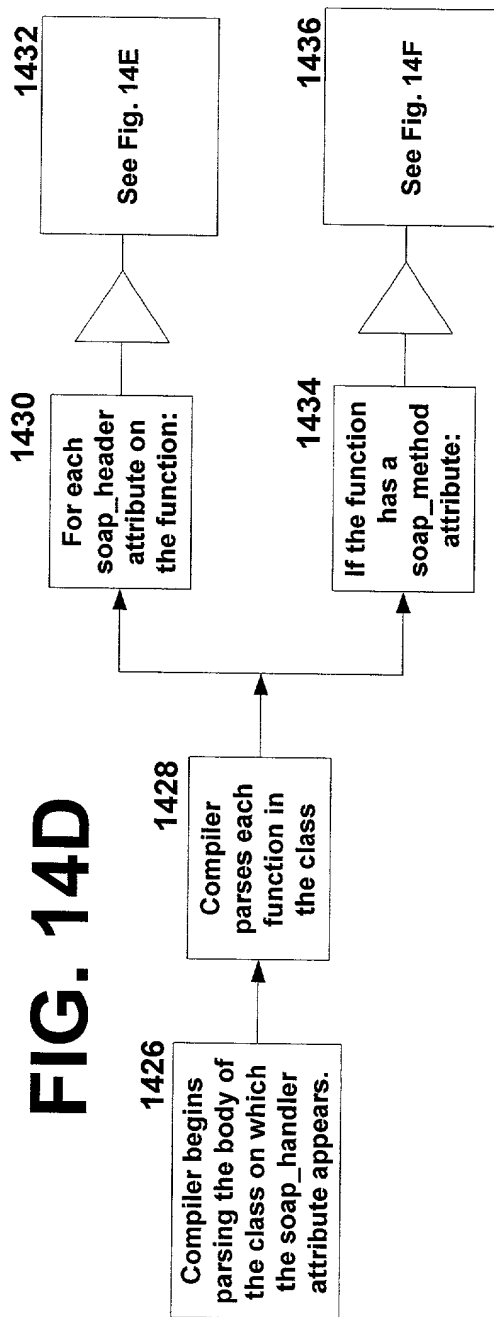


FIG. 14B



**FIG. 14C**



**FIG. 14D**

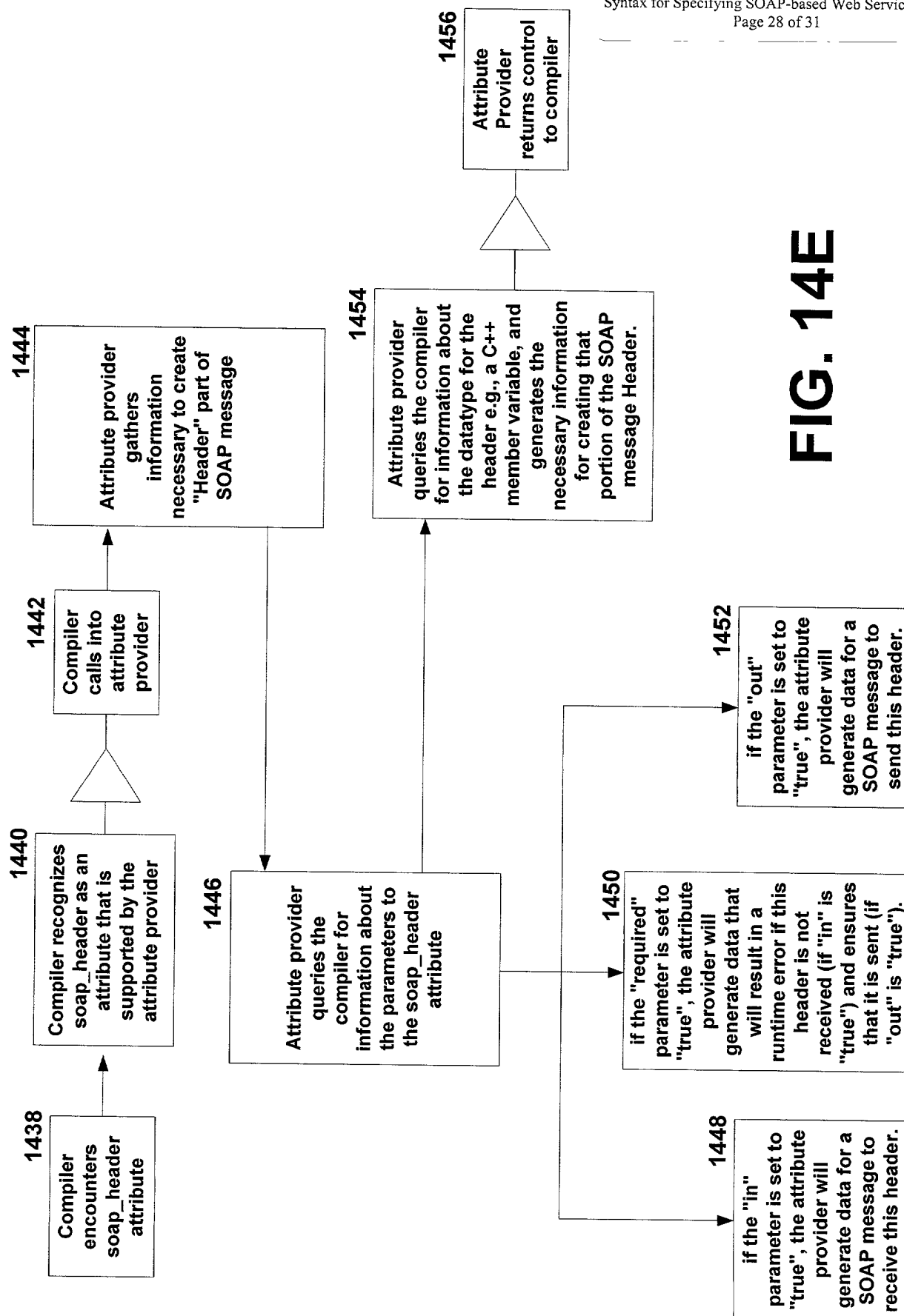


FIG. 14E

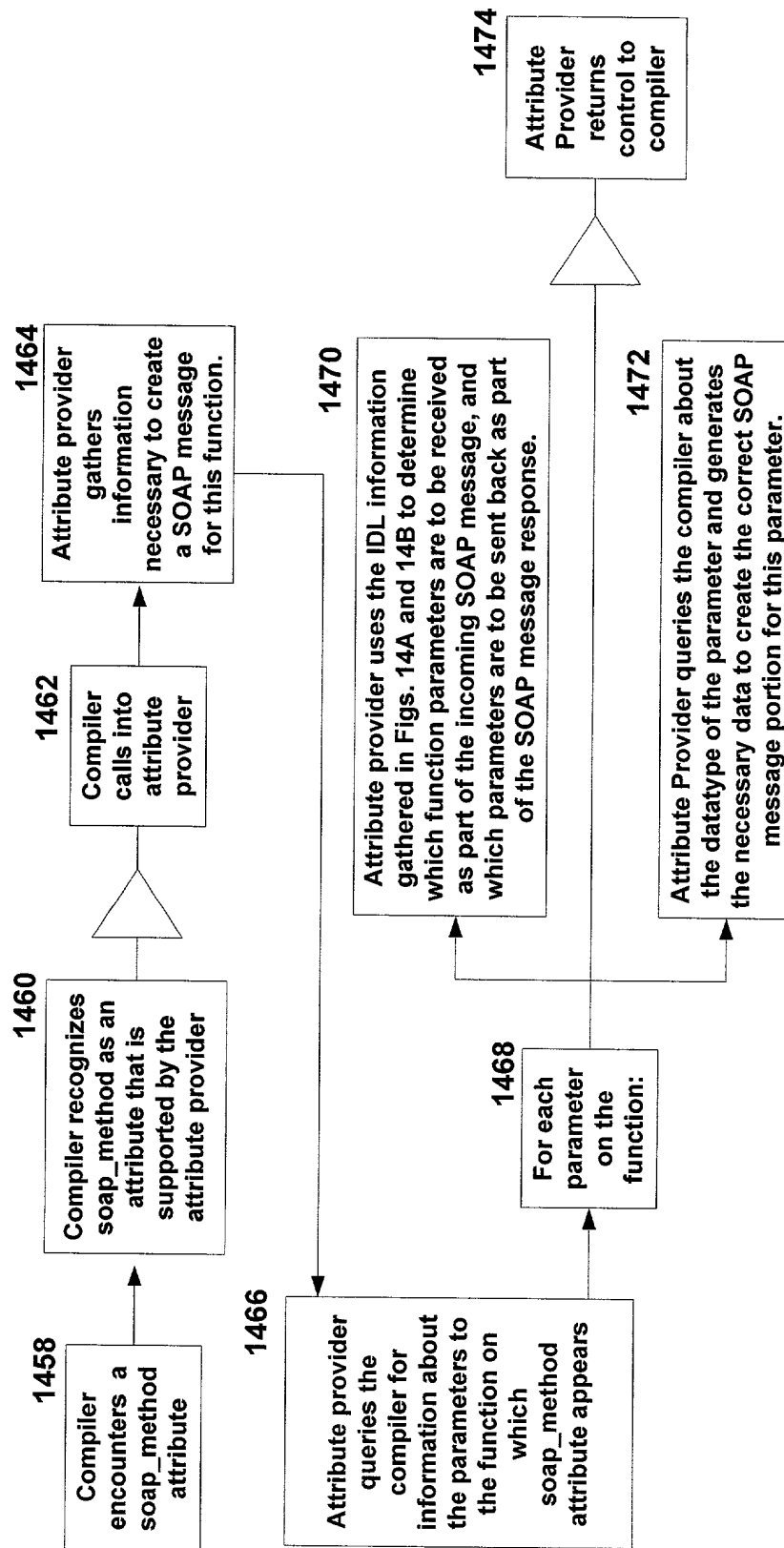


FIG. 14F

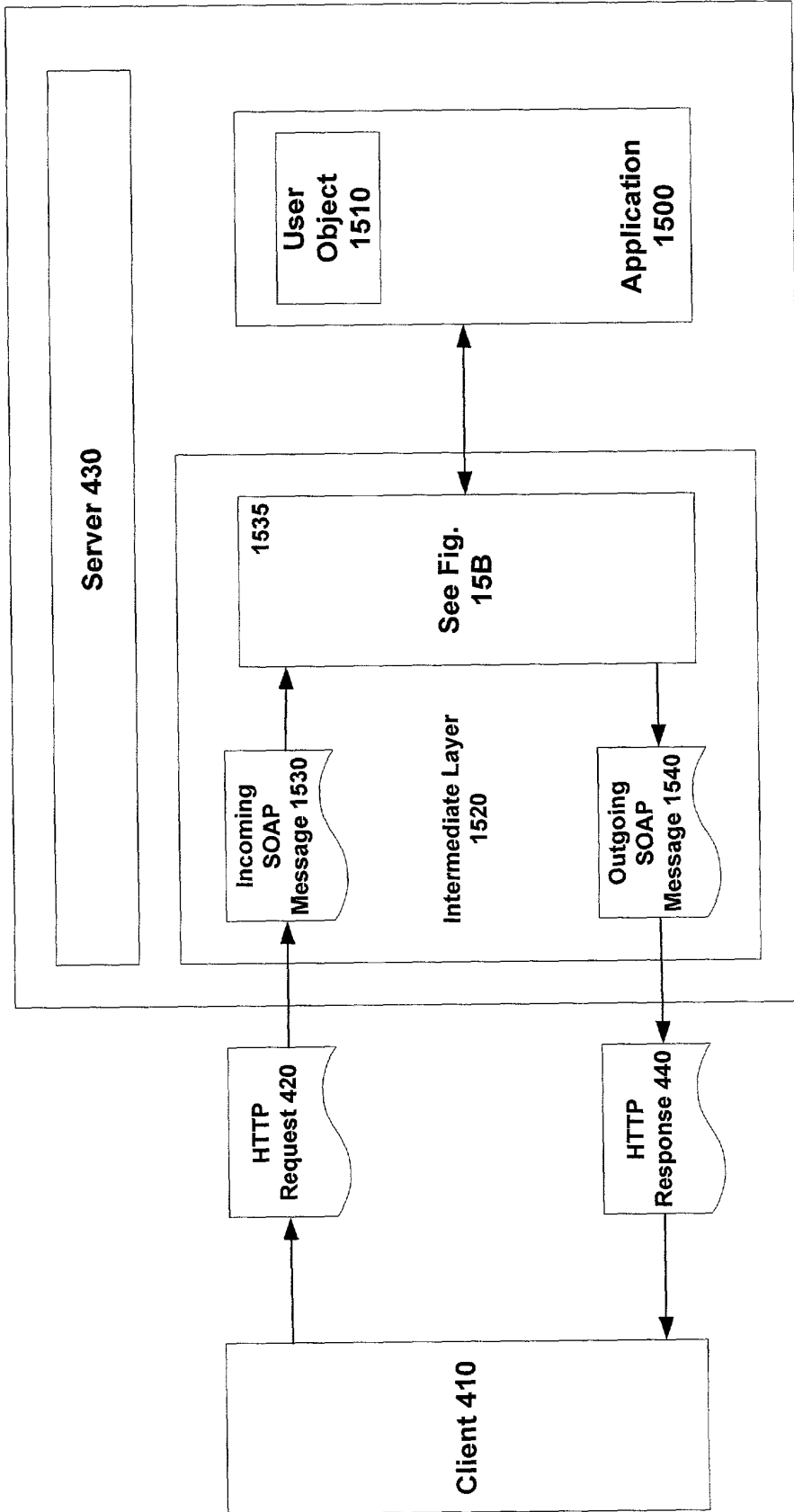
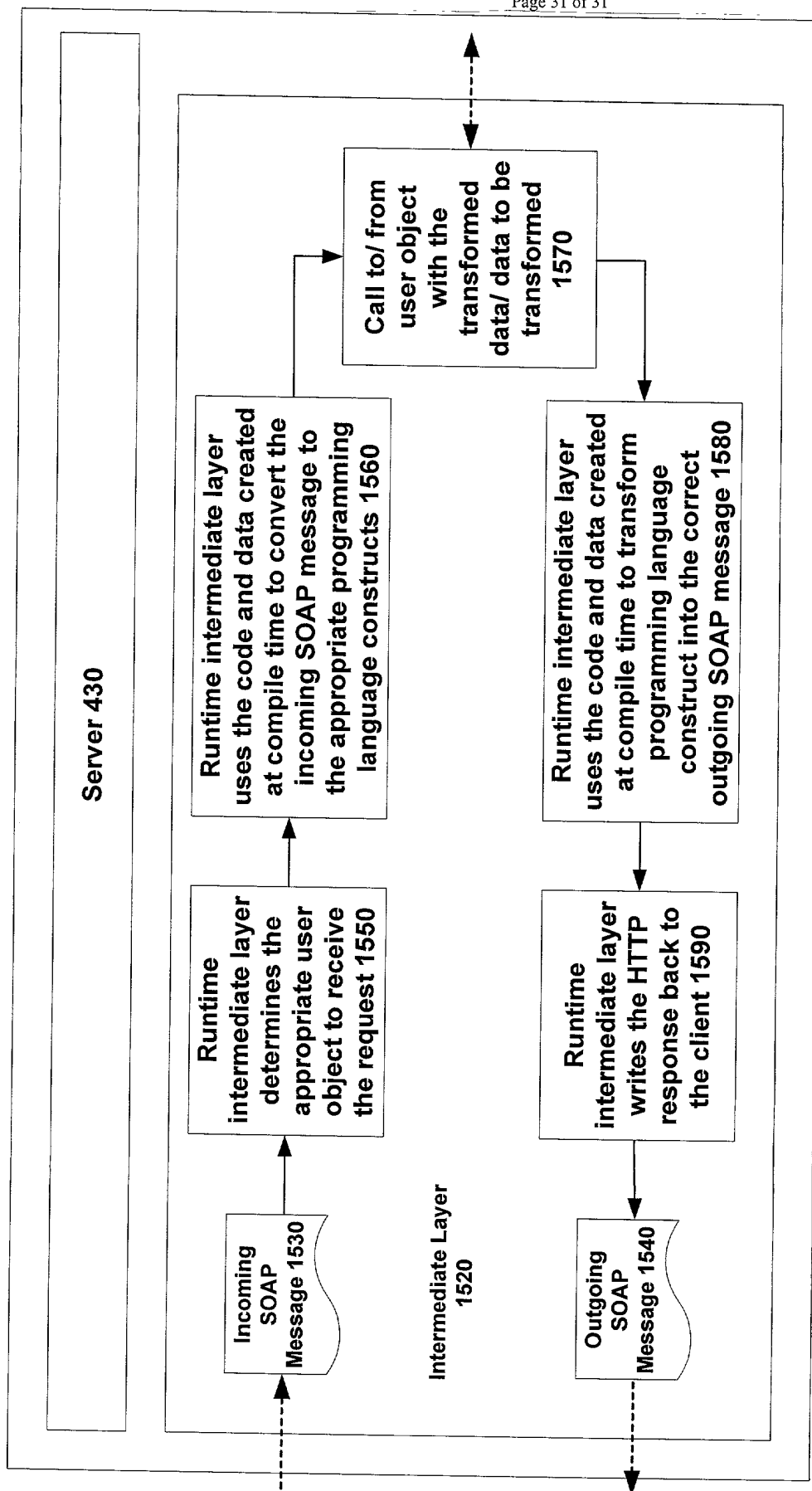


FIG. 15A



**FIG. 15B**